

PROGRAMMABLE LSI AND ITS ARITHMETIC METHOD

Publication number: JP9294069 (A)

Publication date: 1997-11-11

Inventor(s): HIGUCHI TETSUYA; MURAKAWA MASAHIRO

Applicant(s): AGENCY IND SCIENCE TECHN

Classification:

- international: G06F7/00; G06F7/57; G06F7/575; G06F15/18; G06N3/00;
H01L21/82; H03K19/173; G06F7/00; G06F7/48; G06F15/18;
G06N3/00; H01L21/70; H03K19/173; (IPC1-7): G06F7/00;
H03K19/173; G06F15/18; H01L21/82; H03K19/173

- European:

Application number: JP19970027797 19970212

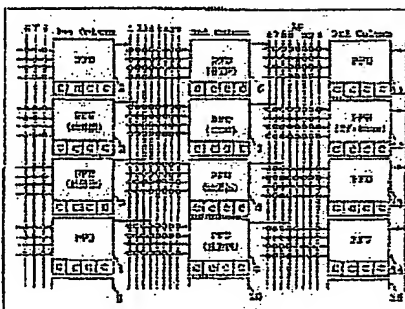
Priority number(s): JP19970027797 19970212; JP19960045223 19960301

Also published as:

JP3170599 (B2)

Abstract of JP 9294069 (A)

PROBLEM TO BE SOLVED: To program a numerical arithmetic expression by designating the type of an arithmetic circuit included in an arithmetic unit based on a desired arithmetic expression and connecting the arithmetic circuit via a connector means. **SOLUTION:** Each of function units PFU 1 to 15 has plural arithmetic circuits of different arithmetic contents. An arithmetical operation circuit, etc., can be constituted by a memory which contains about 10 types of function tables for addition, subtraction, multiplication, subtraction, SIN subtraction, COS subtraction, etc. In such a constitution, the arithmetic result is outputted to the data line of the memory from a designated area when the data designating a desired function table and the data on the arithmetic object are inputted to the address line of the memory. Furthermore, the logical arithmetic tables of AND, OR, etc., are prepared together with an arithmetic circuit which carries out a logical operation of an IF-THEN form. Then the executing arithmetic circuit is selected for every memory and computing element by a switch of a multiplexer, etc., and based on the digital signal that is externally pointed.



Data supplied from the esp@cenet database — Worldwide

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平9-294069

(43) 公開日 平成9年(1997)11月11日

(51) Int.Cl. ⁶	識別記号	片内整理番号	F I	技術表示箇所
H 0 3 K 19/173	1 0 1		H 0 3 K 19/173	1 0 1
G 0 6 F 15/18	5 5 0		G 0 6 F 15/18	5 5 0 C
H 0 1 L 21/82			H 0 1 L 21/82	A
// G 0 6 F 7/00			G 0 6 F 7/00	E

審査請求 有 請求項の数20 O L (全 12 頁)

(21) 出願番号 特願平9-27797

(22) 出願日 平成9年(1997)2月12日

(31) 優先権主張番号 特願平8-45223

(32) 優先日 平8(1996)3月1日

(33) 優先権主張国 日本 (J P)

(71) 出願人 000001144

工業技術院長

東京都千代田区霞が関1丁目3番1号

(72) 発明者 樋口 哲也

茨城県つくば市梅園1丁目1番4 工業技術院電子技術総合研究所内

(72) 発明者 村川 正宏

埼玉県南埼玉郡白岡町新白岡3-4-5

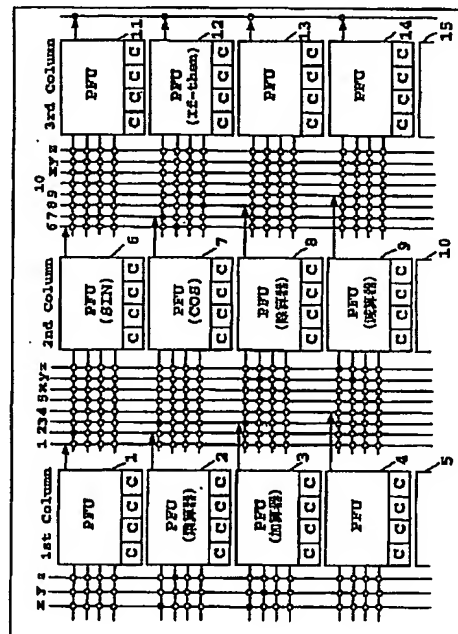
(74) 指定代理人 工業技術院電子技術総合研究所長

(54) 【発明の名称】 プログラマブルLSIおよびその演算方法

(57) 【要約】

【課題】 数値演算式をプログラマブルなLSIを提供する。

【解決手段】 異なる演算回路複数を切り換え的に選択する関数ユニット1~15を用意し、各関数ユニットに対して演算式に対応する四則演算等の種類を指定する。また、演算式の演算の順序に従って関数ユニットをクロスバースイッチにより接続する。



【特許請求の範囲】

【請求項1】 種類の異なる演算回路複数をそれぞれが有し、演算の実行に供する演算回路の指定を外部から受け付け、演算に使用するデータを入力し、演算結果を出力する複数の演算ユニットと、前記複数の演算ユニットを予め定めた演算式に従って相互に接続するための接続手段とを具えたことを特徴とするプログラマブルLSI。

【請求項2】 請求項1に記載のプログラマブルLSIにおいて、前記接続手段はクロスバースイッチであることを特徴とするプログラマブルLSI。

【請求項3】 請求項1に記載のプログラマブルLSIにおいて、前記複数の演算回路の中には数値演算を行う回路を含むことを特徴とするプログラマブルLSI。

【請求項4】 請求項1に記載のプログラマブルLSIにおいて、前記複数の演算回路の中には論理演算を行う回路を含むことを特徴とするプログラマブルLSI。

【請求項5】 請求項1に記載のプログラマブルLSIにおいて、前記演算ユニットは外部から与えられる演算の内容を指示する命令を記憶しておくメモリを有し、該メモリに記憶された命令の指示する演算の内容に対応して、前記複数の演算回路を選択することを特徴とするプログラマブルLSI。

【請求項6】 請求項5に記載のプログラマブルLSIにおいて、前記メモリは複数の前記命令を記憶するための複数の記憶領域を有し、当該複数の記憶領域から順次に前記命令を読み出し、当該読み出した命令に応じて選択された演算回路により演算を行うことを特徴とするプログラマブルLSI。

【請求項7】 請求項1に記載のプログラマブルLSIにおいて、前記接続手段は前記複数の演算ユニットをマトリクス形態で接続可能とすることを特徴とするプログラマブルLSI。

【請求項8】 請求項1に記載のプログラマブルLSIにおいて、前記複数の演算ユニットをツリー構造で接続可能とすることを特徴とするプログラマブルLSI。

【請求項9】 請求項8に記載のプログラマブルLSIにおいて、前記接続手段は前記ツリー構造の階層数を可変設定することにより前記複数の演算ユニットを選択的に接続することを特徴とするプログラマブルLSI。

【請求項10】 請求項9に記載のプログラマブルLSIにおいて、前記接続手段は前記ツリー構造の各階層上に位置し、上または下の階層で隣接する特定の演算ユニットからの演算結果を選択する選択回路を有し、該選択回路により演算結果を選択することにより前記ツリー構造の階層数を可変設定することを特徴とするプログラマブルLSI。

【請求項11】 種類の異なる演算回路複数をそれぞれが有し、演算の実行に供する演算回路の指定を外部から受け付け、演算に使用するデータを入力し、演算結果を

出力する複数の演算ユニットと、前記複数の演算ユニットを予め定めた演算式に従って相互に接続するための接続回路とをLSI化し、前記複数の演算ユニットで実行させる演算回路および前記クロスバースイッチにより接続する演算ユニットを予め定めた演算式に従って指示することを特徴とするプログラマブルLSIの演算方法。

【請求項12】 請求項11に記載のプログラマブルLSIの演算方法において各前記演算ユニットに与える演算回路の種類を示す初期データを順次に変更することにより前記演算式の内容を順次に変更することを特徴とするプログラマブルLSIの演算方法。

【請求項13】 請求項11に記載のプログラマブルLSIの演算方法において、遺伝的アルゴリズムの手法を使用して好適な演算式の内容を決定することを特徴とするプログラマブルLSIの演算方法。

【請求項14】 請求項11に記載のプログラマブルLSIの演算方法において、前記演算ユニットのそれぞれは演算中および演算終了を示すフラグ情報を演算状態に応じてセットし、該演算ユニットに接続する下流側の演算ユニットは前記フラグに基づき接続の上流側の演算結果を取り込むことを特徴とするプログラマブルLSIの演算方法。

【請求項15】 請求項11に記載のプログラマブルLSIの演算方法において、前記複数の演算回路ユニットの各々は演算の内容を指示する命令を記憶しておくメモリを有し、該メモリに記憶された命令の指示する演算の内容に対応して、前記複数の演算回路を選択することを特徴とするプログラマブルLSIの演算方法。

【請求項16】 請求項11に記載のプログラマブルLSIの演算方法において、前記メモリは複数の前記命令を記憶するための複数の記憶領域を有し、当該複数の記憶領域から順次に前記命令を読み出し、選択された演算器により演算を行うことを特徴とするプログラマブルLSIの演算方法。

【請求項17】 請求項11に記載のプログラマブルLSIの演算方法において、前記接続回路は前記複数の演算ユニットをマトリクス形態で接続可能とすることを特徴とするプログラマブルLSIの演算方法。

【請求項18】 請求項11に記載のプログラマブルLSIの演算方法において、前記複数の演算ユニットをツリー構造で接続可能とすることを特徴とするプログラマブルLSIの演算方法。

【請求項19】 請求項18に記載のプログラマブルLSIの演算方法において、前記接続回路は前記ツリー構造の階層数を可変設定することにより前記複数の演算回路を選択的に接続することを特徴とするプログラマブルLSIの演算方法。

【請求項20】 請求項19に記載のプログラマブルLSIにおいて、前記接続手段は前記ツリー構造の各階層上に位置し、上または下の階層で隣接する特定の演算回

路からの演算結果を選択する選択回路を有し、該選択回路により演算結果を選択することにより前記ツリー構造の階層数を可変設定することを特徴とするプログラマブルLSIの演算方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は演算式をプログラマブル（プログラム可能）なプログラマブルLSI（大規模集積回路）およびその演算方法に関する。

【0002】

【従来の技術】従来この種LSIとしてはFPGA（Field Programmable Gate Array）がよく知られている。FPGAはプログラム可能な複数の論理ユニットとこれらユニットを選択的に接続するクロスバースイッチから構成されている。論理ユニットはアンド（AND）、オア（OR）等の論理演算を行う。

【0003】なお、四則演算などを行う回路としてはCPU（Central Processing Unit）、DSP（Digital Signal Processor）もしくは演算式に従って加減算器を組み合わせて専用の固定接続したLSIが知られている。

【0004】

【発明が解決しようとする課題】上記従来例の中でFPGAは論理演算がよく行なわれ、数値演算は難しい。一方、CPUやDSPは数値演算は可能であるものの、数値演算を規定した演算式はプログラムで規定されるので、演算式を変更するためにはプログラム全体を書き換えるという手間が必要となる。また、加減算等の演算器を組み合わせたLSIは演算式の変更ができない、すなわち、プログラマブルではないという欠点を有する。

【0005】そこで、本発明の目的は、上述の欠点に鑑みて、数値演算式をプログラムするのに好適なプログラマブルLSIおよびその演算方法を提供することにある。

【0006】

【課題を解決するための手段】このような目的を達成するために、請求項1の発明は種類の異なる演算回路複数のそれぞれが有し、演算の実行に供する演算回路の指定を外部から受け付け、演算に使用するデータを入力し、演算結果を出力する複数の演算ユニットと、前記複数の演算ユニットを予め定めた演算式に従って相互に接続するための接続手段とを具えたことを特徴とする。

【0007】請求項2の発明は、請求項1に記載のプログラマブルLSIにおいて、前記接続手段はクロスバースイッチであることを特徴とする。

【0008】請求項3の発明は、請求項1に記載のプログラマブルLSIにおいて、前記複数の演算回路の中には数値演算を行う回路を含むことを特徴とする。

【0009】請求項4の発明は、請求項1に記載のプロ

グラマブルLSIにおいて、前記複数の演算回路の中には論理演算を行う回路を含むことを特徴とする。

【0010】請求項5の発明は、請求項1に記載のプログラマブルLSIにおいて、前記演算ユニットは外部から与えられる演算の内容を指示する命令を記憶しておくメモリを有し、該メモリに記憶された命令の指示する演算の内容に対応して、前記複数の演算回路を選択することを特徴とする。

【0011】請求項6の発明は、請求項5に記載のプログラマブルLSIにおいて、前記メモリは複数の前記命令を記憶するための複数の記憶領域を有し、当該複数の記憶領域から順次に前記命令を読み出し、当該読み出した命令に応じて選択された演算回路により演算を行うことを特徴とする。

【0012】請求項7の発明は、請求項1に記載のプログラマブルLSIにおいて、前記接続手段は前記複数の演算ユニットをマトリクス形態で接続可能とすることを特徴とする。

【0013】請求項8の発明は、請求項1に記載のプログラマブルLSIにおいて、前記複数の演算ユニットをツリー構造で接続可能とすることを特徴とする。

【0014】請求項9の発明は、請求項8に記載のプログラマブルLSIにおいて、前記接続手段は前記ツリー構造の階層数を可変設定することにより前記複数の演算ユニットを選択的に接続することを特徴とする。

【0015】請求項10の発明は、請求項9に記載のプログラマブルLSIにおいて、前記接続手段は前記ツリー構造の各階層上に位置し、上または下の階層で隣接する特定の演算ユニットからの演算結果を選択する選択回路を有し、該選択回路により演算結果を選択することにより前記ツリー構造の階層数を可変設定することを特徴とする。

【0016】請求項11の発明は、種類の異なる演算回路複数のそれぞれが有し、演算の実行に供する演算回路の指定を外部から受け付け、演算に使用するデータを入力し、演算結果を出力する複数の演算ユニットと、前記複数の演算ユニットを予め定めた演算式に従って相互に接続するための接続回路とをLSI化し、前記複数の演算ユニットで実行させる演算回路および前記クロスバースイッチにより接続する演算ユニットを予め定めた演算式に従って指示することを特徴とする。

【0017】請求項12の発明は、請求項11に記載のプログラマブルLSIの演算方法において、最上流の前記演算ユニットに与える初期データを順次に変更することにより予め定めた演算式の解を取得することを特徴とする。

【0018】請求項13の発明は、前記解を取得するために遺伝的アルゴリズムの手法を使用することを特徴とする。

【0019】請求項14の発明は、請求項11に記載の

プログラマブルLSIの演算方法において、前記演算ユニットのそれぞれは演算中および演算終了を示すフラグ情報を演算状態に応じてセットし、該演算ユニットに接続する下流側の演算ユニットは前記フラグに基づき接続の上流側の演算結果を読み取ることを特徴とする。

【0020】請求項15の発明は、請求項11に記載のプログラマブルLSIの演算方法において、前記複数の演算回ユニットの各々は演算の内容を指示する命令を記憶しておくメモリを有し、該メモリに記憶された命令の指示する演算の内容に対応して、前記複数の演算回路を選択することを特徴とする。

【0021】請求項16の発明は、請求項11に記載のプログラマブルLSIの演算方法において、前記メモリは複数の前記命令を記憶するための複数の記憶領域を有し、当該複数の記憶領域から順次に前記命令を読み出し、選択された演算器により演算を行うことを特徴とする。

【0022】請求項17の発明は、請求項11に記載のプログラマブルLSIの演算方法において、前記接続回路は前記複数の演算ユニットをマトリクス形態で接続可能とすることを特徴とする。

【0023】請求項18の発明は、請求項11に記載のプログラマブルLSIの演算方法において、前記複数の演算ユニットをツリー構造で接続可能とすることを特徴とする。

【0024】請求項19の発明は、請求項18に記載のプログラマブルLSIの演算方法において、前記接続回路は前記ツリー構造の階層数を可変設定することにより前記複数の演算回路を選択的に接続することを特徴とする。

【0025】請求項20の発明は、請求項19に記載のプログラマブルLSIの演算方法において、前記接続手段は前記ツリー構造の各階層上に位置し、上または下の階層で隣接する特定の演算回路からの演算結果を選択する選択回路を有し、該選択回路により演算結果を選択することにより前記ツリー構造の階層数を可変設定することを特徴とする。

【0026】

【発明の実施の形態】以下、図面を参照して本発明の実施の形態を詳細に説明する。図1は本実施の形態のLSI

$$\text{IF } (\cos(X+Y) > \sin(X*Z')) \text{ then} \\ Z' - Y \text{ else } (X+Y)/Y$$

を示す。この式は $\cos(X+Y)$ の値が $\sin(X*Z')$ の値よりも大きいときには $Z' - Y$ の値を演算結果とし、そうでない場合には $(X+Y)/Y$ の値を演算結果とすることを意味する。

【0031】このような演算式を従来の演算器の組み合わせで示した例を図2に示す。従来では、図2に示す回路をLSI化すると、演算式をを変更することは容易ではないが本実施の形態では、自由に演算式を組み替える

I化する回路の基本構成を示す。図1において1~15は関数ユニット(PFU)である。各関数ユニット(本発明の演算ユニット)は同一のものを使用できる。関数ユニットの各々は演算内容が異なる複数の演算回路を有している。四則演算回路等などは、メモリで構成することができる。メモリ内には加算用、減算用、乗算用、減算用、SINの演算用、COSの演算用等約10種類の関数テーブルが格納されている。

【0027】加算用の関数テーブルはたとえば5+1の場合、数値5、1により定まるメモリ領域に加算結果の数値6が格納されている。したがって、使用したい関数テーブルを指定するデータと、演算対象のデータ(上記加算の場合、数値5、1)をメモリのアドレス線に入力すると、そのアドレスで指定されたメモリ領域から演算の結果がメモリのデータ線に出力される。上記各PFUは上述の演算テーブルの他、定数テーブルをも格納しており、定数のみの出力が可能である。また、AND、OR等の論理演算テーブルやIF THEN形式の論理演算を実行する演算回路をも有している。これらのメモリ、演算器は外部から指示されるデジタル信号によりマルチプレクサ等の切替器により実行に供する演算回路が選択される。なお、本実施の形態ではメモリを使用する演算回路を例としたがその他の形態の演算回路を使用してもよいこと勿論である。

【0028】各PFUはクロスバースイッチ(図中○印で図示)により選択的に相互接続される。なお●で示した箇所は接続を表し、たとえば、第1列目(1st Column)に位置するPFU2は入力データxとyに接続され、これらのデータを入力することを示している。第2列目(2nd Column)のPFU6は第1列目のPFU2と接続し、PFU2の演算出力を入力することを示している。

【0029】したがって、ユーザは各PFUに実行させる演算の種類を不図示のディップスイッチや他の指示装置を用いて指定し、クロスバースイッチを操作することで任意の演算式を設定することができる。ちなみに図1の接続例で設定された演算式は

【0030】

【数1】

ことができる点に注意されたい。

【0032】このようにプログラムされた演算式を実行する時間は次のとおりとなる。sin計算に5ユニットタイム、cos計算に5ユニットタイム、減算に1ユニットタイム、if then計算に2ユニットタイム、乗算に2ユニットタイム、除算に3ユニットタイム要するとすると、各PFUで行われる演算のタイミングと、全体の演算に要する時間は図3に示すようになる。図3

に示すように本実施の形態では演算式を構成する部分演算を並列的に実行できるので、CPUのようにシリアル的に部分演算を実行していく演算装置よりも本実施の形態の方が処理速度が速い。

【0033】次にプログラマブルLSIの具体的な構成例を図9を使用して説明する。

【0034】図9においてPFUブロック100は15組みのPFUを有するチップである。PFUコントローラ200およびシステムコントローラ300がPFUブロック100と対で用意される。システムコントローラ300は外部からチップセレクト信号(CS信号と略記する)、リード信号(RD信号と略記する)またはライト信号(WR信号と略記する)およびシステムアドレス信号を入力し、CS信号により選択されたPFUブロック100に対してRD信号またはWR信号を供給する。RD信号はPFUブロック内のPFUインストラクションメモリやPFU内の後述のレジスタのアドレスを指定する信号であり、RD信号の発生時にSystem addr信号によりアドレスが指定されたインストラクションメモリ(各PFU内に設置)の記憶領域(複数)や上記レジスタから、データが読み出され、System Data信号線に出力される。読み出されるデータは通常は、他のチップ(PFUブロック)に引き渡すデータである。

【0035】逆にWD信号の発生時にはSystem Addr信号によりアドレスが指定されたインストラクションメモリの記憶領域や上記レジスタに対して、実行プログラム命令や演算データ(オペランド)が書き込まれる。インストラクションメモリ116は各PFUについての実行プログラム命令をおよびデータを記憶する。実行プログラム命令には、演算命令、制御命令、即値命令の3種の命令がある。演算命令は本実施の形態では加減乗除など8種の演算を行うための関数が定義されている。

【0036】制御命令は特定のレジスタに対する演算データや演算結果の入出力の指示や、小数点位置の指定、分岐条件の指定等を行う。即値命令は演算すべきデータである。

【0037】IN0~IN7はPFUブロック100で処理すべき8組の入力データである。

【0038】PFUコントローラ100はSystem data信号により転送される情報を解析して後述の信号を作成し、PFUブロック100内のPFUに対して供給する。この供給信号について説明する。

【0039】func信号は各PFUが演算実行可能な8つの関数のうちのどの関数を選択するかを示す信号で、各PFUに対してfunc信号が送られる。たとえば、図16(e)に示すような3つのPFUが接続されている状態で、左側のPFUに対するfunc信号の内容を書き換えることによりそのPFUは加算から乗算に

演算実行する関数を変更する。変更結果が図16(f)になる。

【0040】PFU go信号は全PFUに対して動作開始を指示する信号である。

【0041】PFU sel信号は動作を許可する信号で各PFUに対して供給され、この信号がセットされていないPFUは動作することができない。

【0042】mux信号はPFUブロック100内の複数のPFUの接続構成を指示する信号である。この信号により複数のPFUの信号線の接続/否接続が制御されて、たとえば、図15の(a)、(b)、(c)、(d)のような各種のツリーの接続構成が構築される。

【0043】PFU done信号は各PFUから送られる実行終了信号である。

【0044】CLK enable信号はCLK(基準クロック信号)から作成された動作タイミング信号であり、この信号に同期してPFUが動作する。

【0045】PFUブロック100内のPFUに関する主要構成を図10に示す。図10において、PFU0、PFU1のそれぞれの演算結果をPFU8に入力し、PFU2、PFU3のそれぞれの演算結果をPFU9に入力するというようにして、図10の形態では4層の階層(ツリー)でPFUが接続されている。さらに図16において左端の各階層の隣接の特定のPFU(PFU0、PFU8、PFU12、PFU14)の演算結果を取り出すことができる。この取り出した演算結果中からマルチプレクサ(MPXと略記、セレクトとも呼ばれる)101により所望のものを取り出す。この実施の形態では2つのPFUの間の信号線を接続/断を行うスイッチがない点に注目されたい。

【0046】どのPFUの演算結果をMPX101が取り出すかは上述のmux信号により指示される。たとえば、図15の(a)のようなツリー構造で演算を行いたい場合には、MPX101においてPFU14の演算結果を選択して出力すればよい。図15の(b)のようなツリー構造で演算を行いたい場合にはPFU12の演算結果を選択する。図15(c)のようなツリー構造の場合にはPFU8の演算結果を選択し、図15(d)のツリー構造の場合にはPFU0の演算結果を選択する。

【0047】このような接続構成とすることで、スイッチ群が不要となり、回路構成が簡素化される。また、複数のPFUで所定の演算式を組む場合にも各PFUとそのPFUに割り当てる関数との対応関係を把握することが容易という利点がある。

【0048】上述のPFUの構成の一例を図11に示し、内部構成を示す。

【0049】PFUは主にデータ入力用のFIFOバッファ110、レジスタ群111、乗算器(MPU)114、論理演算ユニット(ALU)115およびPFUインストラクションメモリ116から構成される。上述の

System addr 信号によるアドレス指定により System data 信号の内容が外部から書き込まれる。プログラムカウンタ用のレジスタ PC を介してアドレス制御回路 (Next Address Control) からのアドレス指定により、インストラクションメモリ 116 に格納された命令が順次に読み出され、デコーダ 117 により命令の内容が解析される。この解析結果により、MPU 114、ALU 115 およびレジスタ群 111 が制御され、選択された関数による演算が行われる。

【0050】アドレススタックはサブルーチン (分岐命令) が与えられたときに戻りアドレスを記憶しておくための記憶回路であり、スタックポインタ (SP) の指示するアドレスの戻りアドレスがアドレススタックから読み出される。上述したように PFU は 8 種の関数の演算が可能なので、選択された関数の種類により MPY 114、ALU 115 または MPY 114 および ALU 115 の双方が選択される。

【0051】この演算に関連して、外部入力用の FIFO 110 およびレジスタ群 111 の中の所定のレジスタから演算に使用するデータがバス A-BUS、B-BUS、C-BUS を介して MPY 114 および ALU 115 に転送される。また、MPY 114 の演算結果を ALU 115 に入力することも可能である。上述のレジスタや FIFO からのデータ入力のためにセレクト群 112、113 が使用される。

【0052】この例では MPY 114 は入力 R、S を持ち、演算結果はレジスタ M に格納され、Mout → M0 ~ M2 の経路でレジスタ M0 ~ M2 に MPY 114 の演算結果を格納することが可能である。ALU 115 は入力 U、V を持ち、演算結果はレジスタ A に格納され、レジスタ Aout → A0 ~ A2 の経路でレジスタ A0 ~ A2 に ALU 115 の演算結果を格納することが可能である。これらの演算結果を出力する時にシフトが小数点位置の調整に使用される。レジスタ M、レジスタ A および MPY 114、ALU 115 は共に外部リセット信号あるいはアドレス制御回路からの信号によりレジスタ Mrs、Ars を介してリセットされる。

【0053】レジスタ X、Y は外部入力のデータを格納し、レジスタ c0 ~ c8 は定数を格納する。本実施の形態では前の実施形態で説明した染色体データを格納する。レジスタ im には即値を格納する。

【0054】これらの構成部は CLK enable 信号から作成されたイネーブル信号 (EN の表記を有する信号、たとえば、ENO、ENX 等) により動作可能となる。

【0055】上述の回路の入出力関連の動作説明を図 12、図 13、図 14 を参照して説明する。図 12 は外部から情報入力するときの信号発生タイミングを示す。図 13 は演算結果を出力する場合の信号発生タイミングを

示す。図 14 は PFU の処理開始と停止を行う場合の信号発生タイミングを示す。

【0056】図 12 において、時刻 T1 でのインストラクションメモリ 116 から読み出された命令が入力命令のとき (レジスタ IR の格納命令が入力)、ENX 信号によりレジスタ X 側の FIFO に情報が入力され、時刻 T3 で ENY 信号によりレジスタ Y 側の FIFO に情報が入力される。時刻 T2 での Empty X 信号の発生、時刻 T4 での Empty Y 信号および ENI 信号の発生に応じて 2 つの FIFO 110 からレジスタ X、Y に入力情報に転送される。

【0057】一方、図 13 に示すようにインストラクションメモリ 116 から出力命令が読み出されると (レジスタ IR の格納命令が出力)、CLK enable 信号に同期して EN0 信号が発生し、レジスタ Mout またはレジスタ Aout から演算結果が出力される。

【0058】動作開始にあたっては図 14 に示すように PFU go 信号の発生に応じてプログラムカウンタ用レジスタ PC の値が順次にインCREMENT されて、インストラクションメモリ 116 から順次にプログラム命令が読み出される。PFU go 信号の消去に応じて FIFO reset 信号が発生され、FIFO 110 がリセットされる。また、他の回路も動作を停止する。

【0059】演算に関連する動作タイミングは従来の演算回路と同様であるので、詳細な説明を要しないであろう。

【0060】以上、説明したように上述の実施形態では複数の PFU をツリー構造で接続し、その階層を可変設定可能とすることによりマトリクス形態の PFU の接続構造に比べて、接続構成をより簡素化できる。

【0061】なお、図 10 に示した例では PFU を 15 組み有する PFU ブロックの例を示したがこの例に限定することなく、PFU は所望の組み数とすることができる。上述の実施の形態の他に次の例を実現できる。

【0062】1) 本実施の形態では演算式をプログラムし、演算式に代入する入力データを与えることで演算式の演算結果を取得する例であるが、入力データと演算結果のデータの組を繰り返し与えることで、演算式を学習により変更させることができる。

【0063】このようにして好適な演算式を求める際に遺伝的アルゴリズム (Genetic Algorithm) の手法を使用することができる。このアルゴリズムは本願発明者によりすでに提案されて発表されているが (「進化するハードウェア」 1995 年 BIT (10 月号)) この提案内容を簡単に紹介しておく。

【0064】各演算ユニット内の選択された演算回路の種類と演算ユニット間の接続関係を解とし、この解の候補を、染色体 (0、1 の 2 進ビット列) として表現し、これを複数個用意して初期集団とする。また、これらの解の候補を個体と呼ぶ。たとえば、図 4 に示す曲線上の

黒丸の一つを図5に見られるような0と1の2進ビット列として表現し、これを複数個(図5では4個)用意する。

【0065】解の良さを定義する評価関数を定義し、その値を適応度とする。解く問題ごとに最適な評価関数を用意する。

【0066】適応度の高い個体同士を集団の中から選択する。その選択の方法として一般的なものは比例配分による方法で集団中の適応度の総和において、各個体の占める割合に応じて各個体を選ばれる。図5、図6に割合を示す。

【0067】交差(データの一部分が図7に示すように交互に移動したもの)や突然変異(データの一部分が他のものに変化したもの)を上記選択した2つの個体に適用してさらによい個体を作り出す。このようにして作り出された解の中で評価関数により定まる適応度の低い解は淘汰して解の集団数を一定に保ちつつ、適応度が高くて満足する解が得られるまで上記個体の選択、個体の作り換えを繰り返す。

【0068】この結果、プログラマブルLSIにおいて設定される演算式が自動的に変更され、最終に好適な演算式が得られる。このプログラマブルLSIの用途としてはパターン認識に加えてたとえば、暗号解読や、データの統計の解析、通信におけるデータ圧縮、ATM(交換機)の制御に使用できる。

【0069】2) 通常、加算や乗算など演算の種類に応じて演算時間が異なる。そこで、図1の各PFUでは演算状態を示すフラグ情報を用意して下流側のPFUに演算状態を知らせるとよい。より具体的には、ビット1で演算終了を表し、ビット0で演算中を表すフラグを上流側のPFUは演算開始時および終了時に設定する。このフラグ情報をフリップフロップやラッチ回路に設定する。下流側のPFUでは図8に示すような処理手順でこのフラグ情報を監視し(図8のステップS1→S2のループ処理)、フラグ情報が計算終了を示したときに上流側のPFUからその演算結果を入力する(ステップS3)。また、自己より下流側のPFUのために自己のフラグ情報を0に設定する。指示された種類の演算を実行して演算結果が得られると下流側のPFUへ演算結果を引き渡すための出力用バッファに計算結果をセットして、自己のフラグ情報1にセットする(ステップS4→S5)。このような処理を各PFUが行うことにより非同期でデータを転送することが可能となる。

【0070】

【発明の効果】以上、説明したように請求項1、2、3、11の発明では、演算ユニットの中の演算回路を所望の演算式に従って種類指定し、接続手段(クロスバースイッチ)により接続していくことで、数値演算式をプログラムして行くことができる。

【0071】請求項4の発明では、演算回路の中に論理

回路を含めることで数値演算結果を用いた論理演算式をプログラムすることが可能となる。

【0072】請求項5、15の発明では、各演算ユニットはメモリを有することにより、外部から異なる命令を受け付けることができ、実行する演算内容を固定化せず、変更することができる。

【0073】請求項6、16の発明では、メモリが複数の記憶領域を有することにより、異なる演算内容を指示する命令を一括してメモリに記憶して、順次に異なる演算を行うことができるので、他種、多様の演算式を実行することができ、演算式がプログラマブルとなる。

【0074】請求項7、17の発明では、演算ユニットの接続形態をマトリクス形態とすることで、ありとあらゆる演算を実行することができる。

【0075】請求項8～10、18～20の発明では、演算ユニットの接続形態をツリー構造とすることで、マトリクス形態よりも少ない演算ユニットで演算を実行することができる。また、ツリー構造の階層数を可変することにより、異なる演算式に対応することができる。また、各階層の特定の演算回路の演算結果を選択することで、信号線の接続/断を行うスイッチを特に設ける必要はなく、LSI内部の回路構成を簡素化することができる。もって、システム全体の小型化に寄与することができる。

【0076】請求項12の発明では、演算式の各演算の種類およびその接続を規定する初期データを順次に変更することで、演算式の解すなわち、好適な演算式を取得することができる。これによりプログラマブルLSIでニューラルネットワークや、統計分析回路を構成することができ、パターン認識やデータ解析等を実行することができる。

【0077】請求項13の発明では、取得した解の中で好適な解を選択することができる。

【0078】請求項14の発明では、演算ユニット間のデータ転送を非同期で行うことができる。

【図面の簡単な説明】

【図1】本発明実施の形態の回路構成を示す構成図である。

【図2】演算式に従った演算回路の接続例を示すブロック図である。

【図3】演算処理タイミングを示す説明図である。

【図4】遺伝的アルゴリズムを説明するための説明図である。

【図5】遺伝的アルゴリズムを説明するための説明図である。

【図6】遺伝的アルゴリズムを説明するための説明図である。

【図7】遺伝的アルゴリズムを説明するための説明図である。

【図8】PFU間のデータ通信処理内容を示すフローチ

ャートである。

【図9】他の実施の形態のシステム構成を示すブロック図である。

【図10】PFUの接続構成を示すブロック図である。

【図11】PFUの内部構成を示すブロック図である。

【図12】入力の信号発生タイミングを示すタイミングチャートである。

【図13】出力の信号発生タイミングを示すタイミングチャートである。

【図14】動作開始および停止の信号発生タイミングを示すタイミングチャートである。

【図15】PFUの接続パターン例を示す説明図である。

【図16】PFUの演算内容の切り換えを示す説明図で

ある。

【符号の説明】

1～15 関数ユニット

100 PFUブロック

110 FIFO (バッファ)

111 レジスタ群

112, 113 セレクタ

114 乗算器 (MPU)

115 論理演算ユニット (ALU)

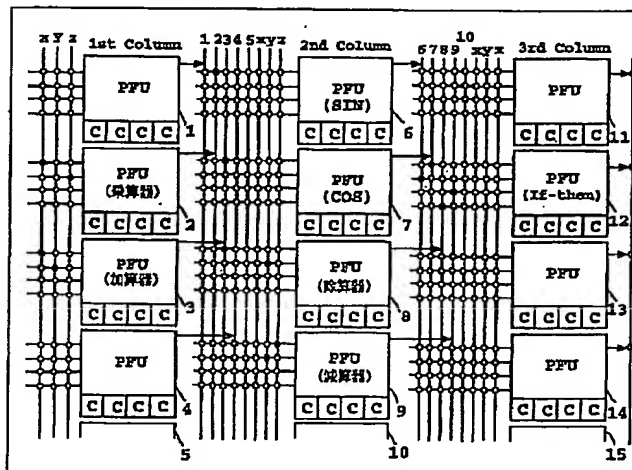
116 インストラクションメモリ

117 デコーダ

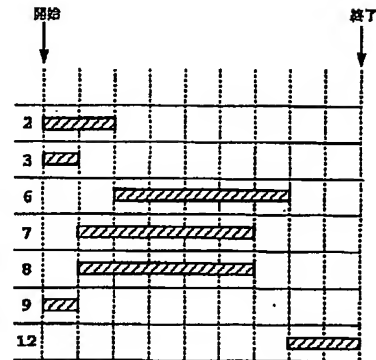
200 PFUコントローラ

300 システムコントローラ

【図1】

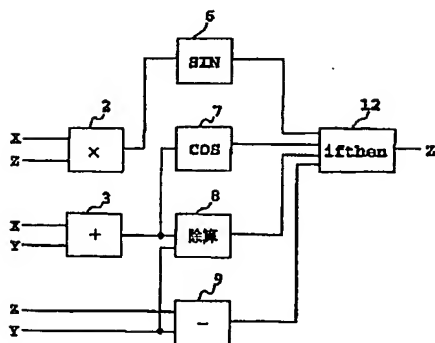


【図3】

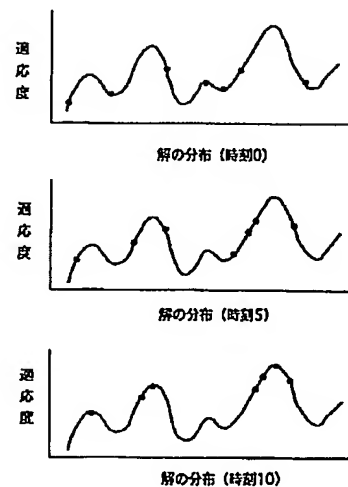
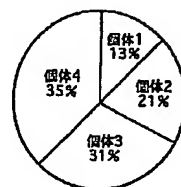


【図4】

【図2】



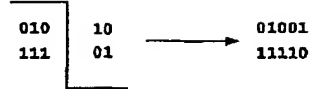
【図6】



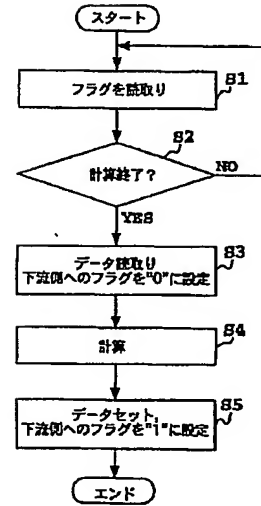
【図5】

値体	適応度	割合 (%)
00100 (4)	2.0	13
01010 (10)	3.2	21
10110 (22)	4.7	31
11101 (29)	5.4	35

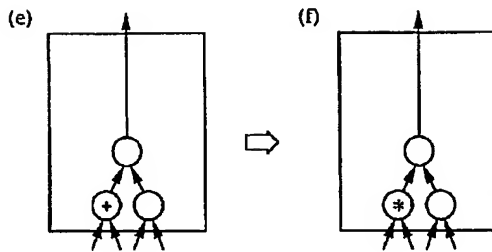
【図7】



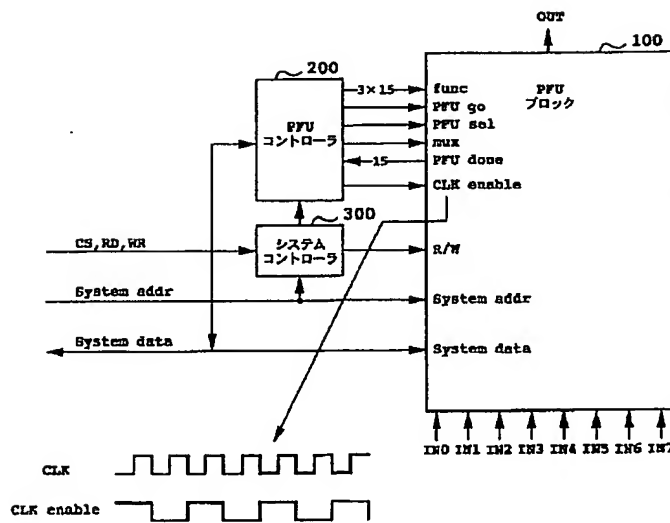
【図8】



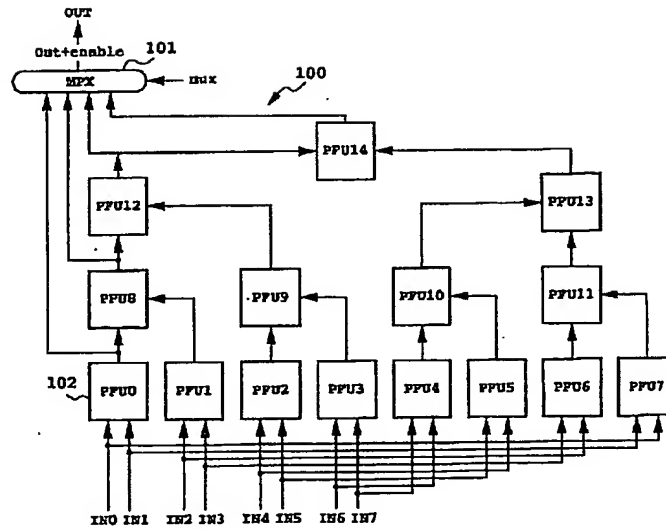
【図16】



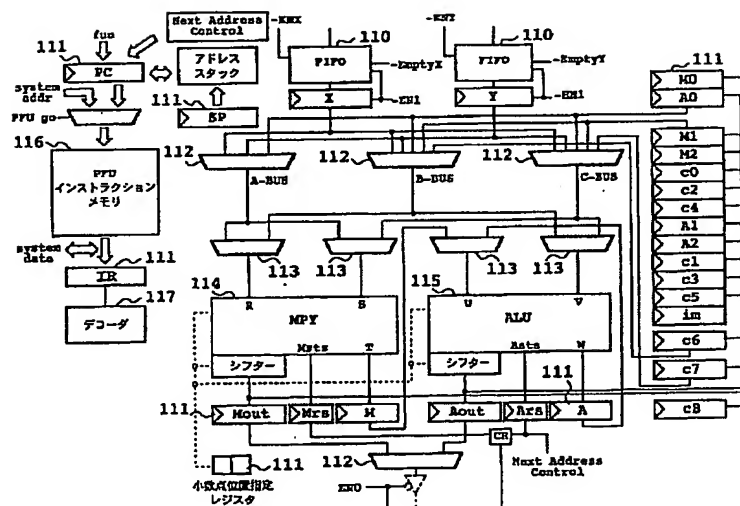
【図9】



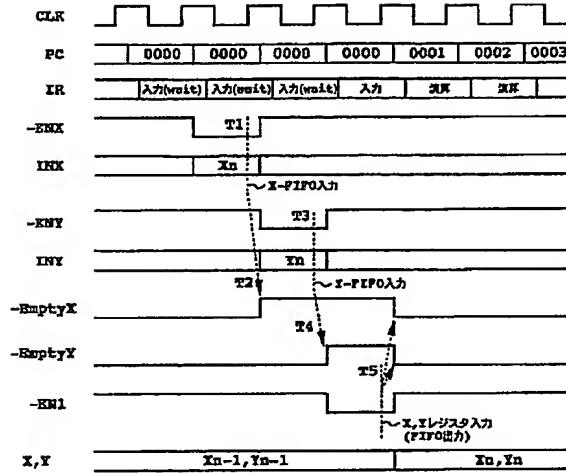
【図10】



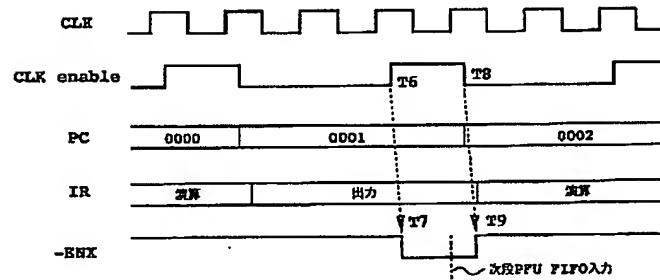
【図11】



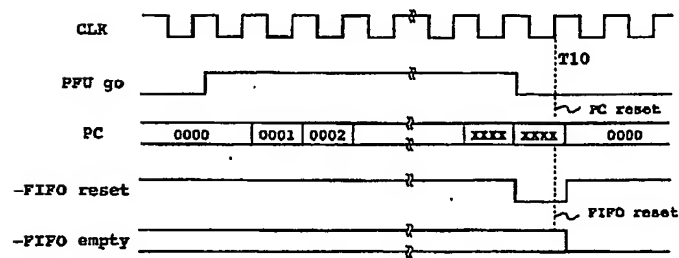
【図12】



【図13】



【図14】



【図15】

